日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２０００年　８月１５日

出願番号
Application Number: 特願２０００－２４６４０４

出願人
Applicant(s): 株式会社ソニー・コンピュータエンタテインメント

CERTIFIED COPY OF
PRIORITY DOCUMENT

２００１年　６月　５日

特許庁長官
Commissioner,
Japan Patent Office 及川耕造

【書類名】　特許願
【整理番号】　SCEI00019
【提出日】　平成12年 8月15日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 11/00
G06F 9/06
【発明の名称】　情報処理システム、実行可能モジュール生成方法および記憶媒体
【請求項の数】　16
【発明者】
【住所又は居所】　東京都中野区中央一丁目３８番１号　住友中野坂上ビル　株式会社シュガーアンドロケッツ内
【氏名】　山本　浩
【発明者】
【住所又は居所】　東京都中野区中央一丁目３８番１号　住友中野坂上ビル　株式会社シュガーアンドロケッツ内
【氏名】　大平　俊充
【特許出願人】
【識別番号】　395015319
【氏名又は名称】　株式会社　ソニー・コンピュータエンタテインメント
【代理人】
【識別番号】　100084032
【弁理士】
【氏名又は名称】　三品　岩男
【電話番号】　045(316)3711
【選任した代理人】
【識別番号】　100087170
【弁理士】
【氏名又は名称】　富田　和子

【電話番号】　045(316)3711

【手数料の表示】

【予納台帳番号】　011992

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9912211

【プルーフの要否】　要

【書類名】明細書

【発明の名称】情報処理システム、実行可能モジュール生成方法および記憶媒体

【特許請求の範囲】

【請求項１】

実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含む、暗号化済みのプロテクト対象オブジェクトを記憶した記憶手段と、

前記記憶手段から前記暗号化済みプロテクト対象オブジェクトを読み出し、当該暗号化済みプロテクト対象オブジェクトを復号する復号手段と、

前記復号後のプロテクト対象オブジェクトと他のオブジェクトとの結合によって作成される実行可能モジュールに前記プロテクトコードを含ませるコード書込み手段と、

前記復号後のプロテクト対象オブジェクトを、前記他のオブジェクトとの結合を行った後に消去する削除手段と、

を有することを特徴とする情報処理装置。

【請求項２】

実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含む、暗号化済みのプロテクト対象オブジェクトを記憶した記憶手段と、

前記記憶手段から前記暗号化済みプロテクト対象オブジェクトを読み出し、当該暗号化済みプロテクト対象オブジェクトを復号する復号手段と、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成するコード生成手段と、

前記復号後のプロテクト対象オブジェクトに前記第一のプロテクトコードを埋め込み、当該第一のプロテクトコード埋込み済みのプログラム対象オブジェクトと他のオブジェクトとの結合により実行可能モジュールが作成されたら、当該実行可能モジュールに、前記第二のプロテクトコードを埋め込むコード書込み手段と、

前記第一のプロテクトコード埋込み済みのプロテクト対象オブジェクトを、前

記二のプロテクトコード埋込み前に消去する削除手段とを有することを特徴とする情報処理装置。

【請求項３】

請求項２記載の情報処理装置であって、

前記コード生成手段は、

乱数から、前記第一のプロテクトコードと前記第二のプロテクトコードを生成することを特徴とする情報処理装置。

【請求項４】

請求項２または３記載の情報処理装置であって、

前記コード書込み手段は、

前記第一のプロテクトコードと前記第二のプロテクトコードとにダミーデータを付加することを特徴とする情報処理装置。

【請求項５】

請求項１、２、３および４のいずれか１項に記載の情報処理装置であって、

前記コード書込み手段は、前記実行可能モジュールに含ませるためのプロテクトコードを暗号化し、

前記プロテクト対象オブジェクトは、前記実行可能モジュールに含まれている暗号化済みのプロテクトコードを、当該プロテクトコードの診断に際して復号する手続きを含む、

ことを特徴とする情報処理装置。

【請求項６】

情報処理装置に処理を実行させるためのプログラムが格納された記憶媒体であって、

前記プログラムは、前記情報処理装置に、

実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成する復号処理と、

前記復号処理で生成されたプロテクト対象オブジェクトと他のオブジェクトとの結合により、前記実行可能モジュールを作成する結合処理と、

前記結合処理による作成される実行可能モジュールに前記プロテクトコードを含ませるためのコード書込み処理と、

前記復号処理で生成されたプロテクト対象オブジェクトを、前記他のオブジェクトとの結合後に消去する削除処理と、

を実行させることを特徴とする記憶媒体。

【請求項７】

情報処理装置に処理を実行させるためのプログラムが格納された記憶媒体であって、

前記プログラムは、前記情報処理装置に、

実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成する復号処理と、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成するコード生成処理と、

前記復号処理後、当該復号処理で生成されたプロテクト対象オブジェクトに、前記第一のプロテクトコードを埋め込む第一コード書込み処理と、

前記コード書込み処理後、当該コード書込み処理で前記第一のプロテクトコードが埋め込まれたプロテクト対象オブジェクトと、他のオブジェクトとの結合により、実行可能モジュールを作成する結合処理と、

前記結合処理後、当該結合処理で作成された前記実行可能モジュールに前記第二プロテクトコードを埋め込む第二コード書込み処理と、

前記第一コード書込み処理後、前記第二コード書込み処理前に、前記復号処理で生成された前記プロテクト対象オブジェクトを消去する削除処理と、

を実行させることを特徴とする記憶媒体。

【請求項８】

請求項７記載の記憶媒体であって、

前記プログラムは、前記コード生成処理において、前記情報処理装置に、

乱数から、前記第一のプロテクトコードと前記第二のプロテクトコードを生成させることを特徴とする記憶媒体。

【請求項９】

請求項７または８記載の記憶媒体であって、

前記プログラムは、前記情報処理装置に、

前記第一のプロテクトコードと前記第二のプロテクトコードとにダミーデータを付加させることを特徴とする記憶媒体。

【請求項１０】

請求項６、７、８および９のいずれか１項に記載の記憶媒体であって、

前記プログラムは、前記情報処理装置に、

前記実行可能モジュールに含ませるためのプロテクトコードを暗号化する処理を実行させ、

前記プロテクト対象オブジェクトには、前記実行可能モジュールに含まれている暗号化プロテクトコードを、当該プロテクトコードの診断に際して復号する手続きが含まれ、

ることを特徴とする記憶媒体。

【請求項１１】

情報処理装置によって処理されるオブジェクトを記憶した記憶媒体であって、

暗号化されたプロテクト対象オブジェクトが格納され、

当該プロテクト対象オブジェクトは、

自身を組み込んだ実行可能モジュールに含まれた１以上のプロテクトコードに矛盾があれば処理を終了させる手続きを含む

ことを特徴とする記憶媒体。

【請求項１２】

請求項１１記載の記憶媒体であって、

前記プロテクト対象オブジェクトは、

前記実行可能モジュールに含まれているプロテクトコードが暗号化されている場合に、当該プロテクトコードの診断に先立ち、当該プロテクトコードを復号する手続きを含む、

ことを特徴とする記憶媒体。

【請求項１３】

複数のオブジェクトを結合して、実行可能モジュールを生成する処理を情報処理装置に実行させる実行可能モジュール生成方法であって、

実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成し、

前記復号後のプロテクト対象オブジェクトと他のオブジェクトとの結合と、前記プロテクトコードの書込みとによって、前記実行可能モジュールを生成し、

前記復号後のプロテクト対象オブジェクトは、前記他のオブジェクトとの結合がなされた後に消去される、

ことを特徴とする実行モジュール生成方法。

【請求項１４】

複数のオブジェクトを結合して、実行可能モジュールを生成する処理を情報処理装置に実行させる実行可能モジュール生成方法であって、

実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成し、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成して、前記復号後のプロテクト対象オブジェクトに当該第一のプロテクトコードを埋め込み、当該第一のプロテクトコード埋込み済みのプログラム対象オブジェクトと他のオブジェクトとの結合により実行可能モジュールを作成したら、当該実行可能モジュールに当該第二のプロテクトコードを埋め込み、

前記第一のプロテクトコード埋込み済みのプロテクト対象オブジェクトは、前記二のプロテクトコード埋込み前に消去されることを特徴とする実行可能モジュール生成方法。

【請求項１５】

複数のオブジェクトの結合により組み立てられた実行可能モジュールを実行可能な装置が実行するための実行可能モジュールが格納された記憶媒体であって、

前記複数のオブジェクトには、

少なくとも１つのプロテクトコードに矛盾があるか否かを診断し、当該診断結

果に応じて、前記実行可能モジュールの処理を終了させる手続きを含んだライブラリオブジェクトが含まれ、

前記実行可能モジュールには、

少なくとも１つのプロテクトコードが埋め込まれている、

ことを特徴とする記憶媒体。

【請求項１６】

複数のオブジェクトの結合によって生成された実行可能モジュールを実行するエンタテインメント装置であって、

前記少なくとも１つのオブジェクトに含まれたプロテクトコードと、それ以外のプロジェクトコードとが前記実行可能モジュールに含まれている場合、それらのプロテクトコード間の関係を診断し、当該関係に矛盾があれば、前記実行可能モジュールの処理を終了させる手段を有することを特徴とするエンタテインメント装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、プログラム開発環境等におけるリソース不正コピーの防止技術に関する。

【０００２】

【従来の技術】

プログラムの不正コピーを防止するための技術として、特許公報第２５６９５６４号記載の、ソフトウエアのコピープロテクト装置が知られている。この装置によれば、フロッピディスクから読み出したプロテクトコードを外部出力しないため、その不正な書替えを防止することができる。これにより、不正に複製されたソフトウエアを実行可能とすることが困難となり、ソフトウエアの不正な複製を効果的に防止することができる。

【０００３】

【発明が解決しようとする課題】

プログラムが作成される場合には、通常、グラフィック機能その他の基本的機

能を提供するライブラリ等のリソースが使用される。このようなライブラリ等のリソースに含まれているオブジェクトは、実行可能モジュールではない。そのため、不正コピー防止のためのプロテクトをライブラリにかけることは困難である。したがって、現在のところ、ライブラリ等のリソースは、メーカ側がプログラム開発者の良識を信頼するというかたちで、プロテクトがかけられていないまま提供されているのが実情である。

【０００４】

しかし、ライブラリ等のリソースは、プログラムと同様、重要な開発成果であるため、なんらかのコピープロテクトがかけられるようにすることが望まれている。

【０００５】

そこで、本発明は、実行可能モジュール以外のオブジェクトの不正コピーを防止することを目的とする。

【０００６】

【課題を解決するための手段】

上記課題を解決するため、本発明は、

実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含む、暗号化済みのプロテクト対象オブジェクトを記憶した記憶手段と、

前記記憶手段から前記暗号化済みプロテクト対象オブジェクトを読み出し、当該暗号化済みプロテクト対象オブジェクトを復号する復号手段と、

前記復号後のプロテクト対象オブジェクトと他のオブジェクトとの結合によって作成される実行可能モジュールに前記プロテクトコードを含ませるコード書込み手段と、

前記復号後のプロテクト対象オブジェクトを、前記他のオブジェクトとの結合を行った後に消去する削除手段と、

を有することを特徴とする情報処理装置を提供する。

【０００７】

また、本発明は、

実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含む、暗号化済みのプロテクト対象オブジェクトを記憶した記憶手段と、

前記記憶手段から前記暗号化済みプロテクト対象オブジェクトを読み出し、当該暗号化済みプロテクト対象オブジェクトを復号する復号手段と、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成するコード生成手段と、

前記復号後のプロテクト対象オブジェクトに前記第一のプロテクトコードを埋め込み、当該第一のプロテクトコード埋込み済みのプログラム対象オブジェクトと他のオブジェクトとの結合により実行可能モジュールが作成されたら、当該実行可能モジュールに、前記第二のプロテクトコードを埋め込むコード書込み手段と、

前記第一のプロテクトコード埋込み済みのプロテクト対象オブジェクトを、前記二のプロテクトコード埋込み前に消去する削除手段とを有することを特徴とする情報処理装置を提供する。

【０００８】

また、これらの情報処理装置によって作成された実行可能モジュールは、記憶媒体に格納することができる。例えば、

複数のオブジェクトの結合により組み立てられた実行可能モジュールを実行可能な装置が実行するための実行可能モジュールが格納された記憶媒体であって、

前記複数のオブジェクトには、

少なくとも１のプロテクトコードに矛盾があるか否かを診断し、当該診断結果に応じて、前記実行可能モジュールの処理を終了させる手続きを含んだライブラリオブジェクトが含まれ、

前記実行可能モジュールには、

少なくとも１のプロテクトコードが埋め込まれている、

ことを特徴とする記憶媒体とすることができる。

【０００９】

ここで、プロテクトコードは、乱数から生成されるようにしてよい。また、プ

ロテクトコードにダミーデータが付加されるようにしてもよい。また、実行可能モジュールに含まれているプロテクトコードを復号する手続きをプロテクト対象オブジェクトに含ませておけば、プロテクトコードを暗号化してから実行可能モジュールに含ませるようにすることもできる。

【００１０】

さらに、本発明は、これらの情報処理装置に処理を実行させるため情報が格納された記憶媒体として、

(１)情報処理装置に処理を実行させるためのプログラムが格納された記憶媒体であって、

前記プログラムは、前記情報処理装置に、

実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成する復号処理と、

前記復号処理で生成されたプロテクト対象オブジェクトと他のオブジェクトとの結合により、前記実行可能モジュールを作成する結合処理と、

前記結合処理による作成される実行可能モジュールに前記プロテクトコードを含ませるためのコード書込み処理と、

前記復号処理で生成されたプロテクト対象オブジェクトを、前記他のオブジェクトとの結合後に消去する削除処理と、

を実行させることを特徴とする記憶媒体、

(２)情報処理装置に処理を実行させるためのプログラムが格納された記憶媒体であって、

前記プログラムは、前記情報処理装置に、

実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含むを、暗号化済みのプロテクト対象オブジェクトの復号により生成する復号処理と、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成するコード生成処理と、

前記復号処理後、当該復号処理で生成されたプロテクト対象オブジェクトに、

前記第一のプロテクトコードを埋め込む第一コード書込み処理と、

前記コード書込み処理後、当該コード書込み処理で前記第一のプロテクトコードが埋め込まれたプロテクト対象オブジェクトと、他のオブジェクトとの結合により、実行可能モジュールを作成する結合処理と、

前記結合処理後、当該結合処理で作成された前記実行可能モジュールに前記第二プロテクトコードを埋め込む第二コード書込み処理と、

前記第一コード書込み処理後、前記第二コード書込み処理前に、前記復号処理で生成された前記プロテクト対象オブジェクトを消去する削除処理と、

を実行させることを特徴とする記憶媒体、

(３)情報処理装置によって処理されるオブジェクトを記憶した記憶媒体であって、

暗号化されたプロテクト対象オブジェクトが格納され、

当該プロテクト対象オブジェクトは、

自身を組み込んだ実行可能モジュールに含まれた１以上のプロテクトコードに矛盾があれば処理を終了させる手続きを含む

ことを特徴とする記憶媒体、

を提供する。

【００１１】

ここで、プロテクト対象オブジェクトは、ライブラリ、画像データ等、実行可能モジュール以外のオブジェクトであってもよい(以下、同じ)。

【００１２】

また、本発明は、

複数のオブジェクトを結合して、実行可能モジュールを生成する処理を情報処理装置に実行させる実行可能モジュール生成方法として、

(１)実行可能モジュールに含まれたプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成し、

前記復号後のプロテクト対象オブジェクトと他のオブジェクトとの結合と、前記プロテクトコードの書込みとによって、前記実行可能モジュールを生成し、

前記復号後のプロテクト対象オブジェクトは、前記他のオブジェクトとの結合がなされた後に消去される、

ことを特徴とする実行モジュール生成方法、および、

(２)実行可能モジュールに含まれた複数のプロテクトコードの矛盾により処理を終了させる手続きを含むプロテクト対象オブジェクトを、暗号化済みのプロテクト対象オブジェクトの復号により生成し、

互いに関連付けられた第一のプロテクトコードと第二のプロテクトコードとを生成して、前記復号後のプロテクト対象オブジェクトに当該第一のプロテクトコードを埋め込み、当該第一のプロテクトコード埋込み済みのプログラム対象オブジェクトと他のオブジェクトとの結合により実行可能モジュールを作成したら、当該実行可能モジュールに当該第二のプロテクトコードを埋め込み、

前記第一のプロテクトコード埋込み済みのプロテクト対象オブジェクトは、前記二のプロテクトコード埋込み前に消去されることを特徴とする実行可能モジュール生成方法を提供する。

【００１３】

この実行可能モジュール生成方法により生成された実行可能モジュールを実行する装置の１つとして、本発明は、

複数のオブジェクトの結合によって生成された実行可能モジュールを実行するエンタテインメント装置であって、

前記少なくとも１つのオブジェクトに含まれたプロテクトコードと、それ以外のプロジェクトコードとが前記実行可能モジュールに含まれている場合、それらのプロテクトコード間の関係を診断し、当該関係に矛盾があれば、前記実行可能モジュールの処理を終了させる手段を有することを特徴とするエンタテインメント装置を提供する。

【００１４】

なお、ここで使用されている各符合(１)(２)(３)は、パラグラフの整理のために付したもの過ぎない。

【００１５】

【発明の実施の形態】

以下、添付の図面を参照しながら、本発明に係る実施の一形態について説明する。ただし、ここでは、ライブラリをプロテクト対象とする。

【００１６】

まず、プログラム開発側で使用される情報処理システムの概略ハードウエア構成について説明する。

【００１７】

プログラム開発側の情報処理システムは、図１に示すように、情報処理装置１００、ユーザ(プログラム開発者)からのデータ入力を受け付ける入力装置(マウス、キーボード等)１２０、各種データを出力する出力装置(ディスプレイ装置等)１３０、等から構成されている。ただし、このシステムには、その他の装置が必要に応じて増設されることもある。

【００１８】

情報処理装置１００は、後述の実行可能プログラムファイル作成処理等を実行するＣＰＵ１０１、グラフィック描画処理等を実行するグラフィックス機構１０２、主記憶１０３、ＣＰＵ１０１と主記憶１０３との間をつないだプロセッサバス１０５、主記憶１０３またはバス１０６とＣＰＵ１０１等との間のデータ転送を制御するメモリ／バス制御チップ１０４、グラフィックス機構１０２のグラフィックス制御チップとメモリ／バス制御チップ１０４との間をつないだＡＧＰ、ソフトウエア開発のために必要となる各種データ(シリアル番号等の識別情報が埋め込まれたフロントエンドツール、フロントエンドツールによって起動されるリンカおよびコンパイラ、フロントエンドツールとともに提供された暗号化ライブラリファイル等)が格納されたハードディスク１０６、ハードディスク１０６に対するデータ読込み／書込み処理を実行するハードディスク・ドライブ１０７、ハードディスクドライブ１０７がつながれた高速バス１０６、記憶媒体Ａに対するデータ読込み読書き／書込み処理を実行するドライブ１１２、入力装置１２０からのデータ転送を制御するコントローラ１１１、出力装置１３０へのデータ転送を制御するコントローラ１１３、各コントローラ１１１，１１３とドライブ１１２とがつながれた低速バス１０９、低速バス１０９と高速バス１０６との間をつないだブリッジ回路１１０、通信ポート(シリアルポート、パラレルポート

等)、等を備えている。なお、ここで示した情報処理装置１００の内部構成は、ソフトウエア開発に使用されるものとしての一例である。

【００１９】

つぎに、この情報処理システムのハードディスク１０６内の格納データについて説明する。

【００２０】

コンパイラは、ＣＰＵに、指定ソースプログラムファイルに記述されたソースの解析処理(字句解析、構文解析、意味解析)等によって、オブジェクトとして翻訳編集させ、それをオブジェクトファイルに格納させるための処理が定義されたソフトウエアである。リンカは、ＣＰＵに、指定オブジェクトファイル群のそれぞれに記述されているオブジェクトを結合編集させることによって実行可能モジュールを組み立させ、それを実行可能プログラムファイルに格納させるための処理が定義されたソフトウエアである。そして、本実施の形態に係るフロントエンドツールは、ＣＰＵに、コンパイラおよびリンカを起動させ、後述の実行可能プログラムファイル作成処理を実行させるための処理が定義されたソフトウエアである。

【００２１】

暗号化Ｌｉｂファイル$Ｌ_1$は、フロントエンドツールとともに、ユーザ(プログラム開発者)に提供される。例えば、暗号化Ｌｉｂファイル$Ｌ_1$が記録されたインストール用ＣＤ－ＲＯＭと、フロントエンドツールが記録されたインストール用フロッピィディスクとがワンセットとしてユーザに提供される。

【００２２】

この暗号化Ｌｉｂファイル$Ｌ_1$は、適当な暗号化処理によって１以上のＬｉｂファイルを暗号化することによって作成されたものである。ユーザは、実行可能プログラムファイル作成時に、この暗号化Ｌｉｂファイル$Ｌ_1$をフロントエンドツールで復号化することによって、実行可能プログラムファイル作成に必要な１以上のＬｉｂファイル$Ｌ_2$が得ることができる。このように、ライブラリを暗号化された状態で提供し、それと共に提供されたフロントエンドツールによってその復号化を行なうようにしたのは、フロントエンドツールを持たないユーザによ

るＬｉｂファイル不正使用を禁じるためである(後述)。

【００２３】

そして、暗号化ＬｉｂファイルＬ$_1$の復号により生成されるＬｉｂファイルＬ$_2$には、グラフィック機能、イベント処理等、アプリケーションにおいて使用される基本的な関数等のオブジェクトが部品化されて格納されている。また、すべてのアプリケーションにおいて呼び出される初期化関数が含まれているＬｉｂファイル内オブジェクトには、さらに、初期化関数において呼び出されるプロテクトコードチェック手続きと、オブジェクトファイルＭ$_2$との結合編集前に第一プロテクトコードＣ$_1$が格納される第一プロテクトコード格納変数とが定義されている。

【００２４】

ここで、プロテクトコードチェック手続きには、第一プロテクトコード格納変数の格納データとプログラムファイル末尾の格納データとが、２つのプロテクトコードＣ$_1$,Ｃ$_2$の満たすべき関係を満たしているか否かをチェックし、その関係を満たさなければ実行中プログラムを初期化段階で終了させる手続きが定義されている。

【００２５】

なお、本実施の形態では、初期化関数が含まれているＬｉｂファイルＬ$_2$だけにプロテクトコードチェック関数および第一プロテクトコード格納変数を含ませているが、すべてのＬｉｂファイルにプロテクトコードチェック関数および第一プロテクト格納変数を含ませることにしてもよいし、代表的なＬｉｂファイル、例えば、プロテクトをかける必要のあるＬｉｂファイルだけにプロテクトコードチェック関数および第一プロテクト格納変数を含ませることにしてもよい。

【００２６】

本実施の形態に係る情報処理システムは、このようなハードウエア構成およびハードディスク格納データ(フロントエンドツール、コンパイラ、リンカ等のソフトウエア、暗号化ライブラリファイル)によって、図２に示すような機能構成を実現する。

【００２７】

具体的には、(１)フロントエンドツールの起動コマンドの入力を受け付ける入力受付部２００、(２)起動時にフロントエンドツール複製チェック処理を実行する起動時チェック処理部２０１、(３)開発成果(ソースプログラムファイルM_1、オブジェクトファイルM_2、実行可能プログラムファイルM_3)を格納するための開発成果記憶部２０２、(４)ソースプログラムファイルM_1からオブジェクトファイルM_2を生成するコンパイル処理部２０３、(５)暗号化ライブラリファイルL_1(暗号化ＬｉｂファイルL_1と呼ぶ)を記憶したライブラリ記憶部２０４、(６)暗号化ＬｉｂファイルL_1を複号する復号化処理部２０５、(７)復号化処理部２０４の復号化処理によって生成されたライブラリファイル(Ｌｉｂファイルと呼ぶ)を一時ファイルL_2として記憶する一時ファイル記憶部(ｔｍｐ)２０６、(８)ＬｉｂファイルL_2とオブジェクトファイルM_2との結合編集により実行可能プログラムファイルM_3を生成するリンク処理部２０７、(９)リンク処理部２０７の結合編集処理後に一時ファイルL_2を削除する一時ファイル削除処理部２０８、(１０)リンク処理部２０７の結合編集処理前のＬｉｂファイルL_2とこのＬｉｂファイルL_2から作成された実行可能プログラムファイルM_3とにそれぞれプロテクトコードC_1,C_2を付与するコード付与処理部２０９、(１１)これら各処理部を制御する制御処理部２１０、が実現される。

【００２８】

なお、コード付与処理部２０９は、(１２)互いに関連付けられた１組のプロテクトコードC_1,C_2(例えば、$C_1=f(C_2)$の関係を満たす１組のプロテクトコードC_1,C_2)を生成するコード生成処理部２０９ａと、(１３)コード生成処理部２０９ａが生成した１組のプロテクトコードC_1,C_2のうちの一方のコードC_1(第一プロテクトコードと呼ぶ)をＬｉｂファイルL_2内のコード格納変数に書き込むとともに他方のコードC_2(第二プロテクトコードと呼ぶ)を実行可能プログラムファイルM_3の末尾に付加するコード書込処理部２０９ｂとからなっている。

【００２９】

つぎに、これら各機能構成部により実行される実行可能プログラムファイル作成処理について説明する。

【００３０】

プログラム開発者は、コーディングが終了したら、ソースプログラムファイルM_1をハードディスク１０８の所定の領域(図２の開発成果記憶部２０２に相当)に格納してから、そのソースプログラムファイル名と必要なライブラリファイル名とを指定してフロントエンドツールを起動する。これにより、ＣＰＵが、フロントエンドツールに定義された処理を実行し、以下の実行可能プログラムファイル作成処理(図２のフローチャート参照)が実現する。

【００３１】

まず、フロントエンドツールの起動コマンドを受け付けた入力受付部２００は、制御処理部２１０に処理開始を指示する(Ｓ３００)。この指示を受け付けると、制御処理部２１０は、フロントエンドツール複製チェック処理の実行を起動時チェック処理部２０１に指示する。

【００３２】

この指示にしたがって、起動時チェック処理部２０１は、以下に示す方法により、フロントエンドツールの不正使用をチェックする(Ｓ３０１)。

【００３３】

例えば、プログラム開発環境にＬＡＮが構築されており、ネットワーク上で起動中のプログラムの識別情報が重複していないか否かを監視するチェックプログラムがネットワーク上に常駐している場合であれば、起動時チェック処理部２０１は、フロントエンドツールに付与されている識別情報をネットワーク上に送信する。これに対して、チェックプログラムは、それと同じ識別情報がネットワーク上に存在するか否かを返信する。なお、フロントエンドツールがコピー品であれば、同じ識別情報が複数ネットワーク上に存在しており、フロントエンドツールがコピー品でなければ、同じ識別情報が複数ネットワーク上に存在することはない。

【００３４】

また、フロントエンドツールにドングルが割り当てられている場合であれば、起動時チェック処理部２０１は、情報処理装置の通信ポートに装着されているドングルに認証メッセージを送る。これに対して、ドングルは、自身に割り当てられた識別情報を返信する。なお、認証されたユーザがフロントエンドツールを起

動した場合であれば、ドングルから返信された識別情報が、フロントエンドツールの識別情報と一致し、認証されたユーザ以外のユーザがフロントエンドツールを起動した場合であれば、ドングルの装着がないために識別情報の返信がないか、または、ドングルから返信された識別情報がフロントエンドツールの識別情報と一致しないことになる。

【００３５】

このようなチェックの結果に基づき、起動時チェック処理部２０１は、フロントエンドツールの使用に不正(フロントエンドツールが複製品である、認証ユーザ以外のユーザによる使用である、等)があるか否かを判断する(Ｓ３０２)。

【００３６】

その結果、起動時チェック処理部２０１がフロントエンドツールの使用に不正があると判断した場合、制御処理部２１０が、エラーメッセージを出力してから(Ｓ３０３)、処理を終了させる(Ｓ３０８)。これにより、フロントエンドツールの不正使用が禁止されるため、フロントエンドツールの不正使用者には、暗号化Ｌｉｂファイルを復号することが困難となる。したがって、Ｌｉｂファイルの不正使用も禁止される。

【００３７】

一方、起動時チェック処理部２０１がフロントエンドツールの使用に不正がないと判断した場合、制御処理部２１０が、指定のソースプログラムファイルM_1の機械語翻訳をコンパイル処理部２０３に指示する。

【００３８】

この指示に応じて、コンパイル処理部２０３は、指定のソースプログラムファイルM_1をアセンブリプログラムにコンパイルし、そのアセンブリプログラムをオブジェクトにアセンブルする。そして、ここで生成されたオブジェクトファイルM_2を開発成果記憶部２０２に格納してから、制御処理部２１０に正常終了を報告する。

【００３９】

制御処理部２１０は、コンパイル処理部２０３から正常終了報告を受け付けると、Ｌｉｂファイルの復号化を復号化処理部２０５に指示する。

【００４０】

この指示に応じて、復号化処理部２０５は、ライブラリ記憶部２０４から暗号化ＬｉｂファイルL$_1$を取り出し、それを復号化する。そして、ここで生成された１以上のＬｉｂファイルL$_2$を、適当なファイル名で、一時ファイル記憶部２０６に一時ファイルとして格納してから、制御処理部２１０に正常終了を報告する。

【００４１】

制御処理部２１０は、復号化処理部２０５からの正常終了報告を受け付けると、ＬｉｂファイルL$_2$に対するプロテクトコード付与をコード付与処理部２０９に指示する。

【００４２】

この指示に応じて、コード付与処理部２０９では、まず、コード生成処理部２０９ａが、乱数Ｃを発生する。そして、図４に示すように、この乱数Ｃを２つの加工方法F$_1$,F$_2$(例えば、乱数Ｃを入力データとする２つの関数等)でそれぞれ加工することによって２つのプロテクトコードC$_1$,C$_2$を生成し、さらに、これらのプロテクトコードC$_1$,C$_2$をそれぞれ適当な暗号化方法によって暗号化する。ついで、コード書込み処理部２０９ｂが、第二プロテクトコードC$_2$を保持するとともに、所定のＬｉｂファイルL$_2$内の第一プロテクトコード格納変数に第一プロテクトコードC$_1$を格納する(Ｓ３０４)。なお、ここでＬｉｂファイルL$_2$に埋め込まれた第一プロテクトコードC$_1$の暗号化前のコードと、コード書込み処理部２０９ｂによって保持されている第二プロテクトコードC$_2$の暗号化前のコードとの間には、前述したように、予め定められた関係(ここでは、C$_1$＝ｆ(C$_2$)とする)がある。

【００４３】

ＬｉｂファイルL$_2$に対するプロテクトコード付与が終了すると、制御処理部２１０は、リンク処理部２０７に、オブジェクトファイルM$_2$と指定のＬｉｂファイルL$_2$との結合編集を指示する。

【００４４】

この指示に応じて、リンク処理部２０７は、オブジェクトファイルM$_2$と指定

のＬｉｂファイルL_2とを結合し、実際の番地の埋込み等を行う。そして、これにより組み立てられた実行可能プログラムファイルM_3を、適当な名前で開発成果記憶部２０２に格納してから、正常終了を制御処理部２１０に報告する(Ｓ３０５)。

【００４５】

制御処理部２１０は、リンク処理部２０７からの正常終了報告を受け付けると、一時ファイル削除を一時ファイル削除処理部２０８に指示する。この指示に応じて、一時ファイル削除処理部２０８は、すべてのＬｉｂファイルL_2を一時ファイル記憶部２０６から削除する(Ｓ３０６)。

【００４６】

その後、制御処理部２１０は、実行可能プログラムファイルL_3に対するプロテクトコード付与をコード付与処理部２０９に指示する。

【００４７】

この指示に応じて、コード付与処理部２０９では、コード書込み処理部２０９ｂが、保持中の第二プロテクトコードC_2を実行可能プログラムファイルM_3に付加する。そして、正常終了を制御処理部２１０に報告する(Ｓ３０７)。なお、ライセンシー管理(例えば、プログラム作成会社の識別等)のために、コード付与処理部２０９が、このとき、さらに、フロントエンドツールの識別情報を実行可能プログラムファイルM_3に埋め込まむようにしてもよい。

【００４８】

制御処理部２１０は、コード付与処理部２０９からの正常終了報告を受け付けると、処理を終了させる(Ｓ３０８)。なお、以上の処理により作成された実行可能プログラムファイルM_3は、例えば、光ディスク等の記憶媒体に焼き付けられて市場に供給される(図９参照)。

【００４９】

本実施の形態に係る実行可能プログラムファイル作成処理によれば、以下に示すように、Ｌｉｂファイルの不正使用が防止される。

(１)ハードディスクに格納されたＬｉｂファイルが暗号化されており、かつ、それを復号するフロントエンドツールの不正使用が、起動時チェック(図２のＳ３

０１)によって禁止される。このため、フロントエンドツールの正当使用者以外のユーザによるＬｉｂファイル使用が防止される。

(２)図５に示すように、実行中のプログラムファイルに埋め込まれたプロテクトコードに矛盾があった場合にその処理を終了させるプロテクトコードチェック手続きＥをＬｉｂファイルL_2にあらかじめ含めておくとともに、フロントエンドツールの使用によってＬｉｂファイルL_2とオブジェクトファイルM_2とが結合された実行可能プログラムファイルM_3にだけ、実行可能プログラムファイル作成処理の過程(Ｓ３０４、Ｓ３０７)において適正なプロテクトコードC_1,C_2が埋め込まれるようにすることによって、フロントエンドツールの正当使用により作成された実行可能プログラムファイルM_3のみを実行可能とすることができる。すなわち、フロントエンドツールによらずＬｉｂファイルが結合された実行可能プログラムファイル、具体的には、不正なプロテクトコード(一方または両方)が埋め込まれた実行可能プログラムファイル、プロテクトコード(一方または両方)が埋め込まれていない実行可能プログラムファイル等の実行が、初期化段階において阻止される。このため、フロントエンドツールの正当使用者以外のユーザによるＬｉｂファイル使用が防止される。

(３) 図５に示すように、乱数Ｃから生成された１組のプロテクトコードC_1,C_2のうち、第一プロテクトコードC_1だけをＬｉｂファイルL_2に埋め込んでおき(Ｓ３０４)、その後、ＬｉｂファイルL_2とオブジェクトファイルM_2との結合編集(Ｓ３０５)が終了したら、実行可能プログラムファイルM_2への第二プロテクトコードC_2の埋込み前にＬｉｂファイルL_2を消去(Ｓ３０６)してしまうため、使用済みＬｉｂファイルL_2の不正利用が防止される。すなわち、一方のプロテクトコードC_1が埋め込まれた使用済みＬｉｂファイルL_2が消去前に取り出されたとしても、乱数Ｃから生成された他方のプロテクトコードC_2の取り出しがその段階(Ｓ３０４～Ｓ３０６)では不可能であるから、その使用済みＬｉｂファイルL_2に埋め込まれている第一プロテクトコードC_1と矛盾しない第二プロテクトコードC_2を知ることは困難である。このため、消去前に取り出した使用済みＬｉｂファイルL_2とオブジェクトファイルとを結合編集したとしても、使用済みＬｉｂファイルL_2に埋込済みの第一プロテクトコードと適正な関係を有する第

二プロテクトコードを実行可能プログラムファイルに埋め込むことは困難である。したがって、使用済みＬｉｂファイルの不正使用が防止される。

(４)最終的に作成された実行可能プログラムファイルM_3の解析によって、それに埋め込まれた１組のプロテクトコードC_1,C_2を知ってから、つぎのフロントエンドツール起動時に使用済みＬｉｂファイルL_2を消去前に取り出したとしても、プロテクトコードC_1,C_2は、毎回、乱数から生成されるため、消去前に取り出した使用済みＬｉｂファイルL_2に埋込済みの第一プロテクトコードC_1に対応する第二プロテクトコードは、実行可能プログラムファイルM_3の解析によって知った第二プロテクトコードC_2とは必ずしも一致していない。このため、最終的に作成された実行可能プログラムファイルM_3の解析によっても、適正な関係を有する１組のプロテクトコードを得ることは困難である。したがって、消去前に取り出した使用済みＬｉｂファイルL_2とオブジェクトファイルとを結合編集したとしても、使用済みＬｉｂファイルL_2に埋込済みの第一プロテクトコードと適正な関係を有する第二プロテクトコードを実行可能プログラムファイルに埋め込むことは困難である。このため、使用済みＬｉｂファイルの不正使用が防止される。

【００５０】

以上の４つの防止策によってＬｉｂファイル不正使用が防止されるが、ライブラリバージョンアップのたびに、Ｌｉｂファイルの暗号化方法とプロテクトコードチェック手続きの内容とをそれぞれ変更するようにすれば、Ｌｉｂファイルの不正使用防止が強化される。なお、このようにする場合には、Ｌｉｂファイル複合方法および乱数加工方法が変更されたフロントエンドツールが、新バージョンの暗号化Ｌｉｂファイルとともにリリースされる必要がある。

【００５１】

ところで、以上においては、Ｌｉｂファイル内L_2で定義された第一プロテクトコード格納変数に第一プロテクトコードC_1を格納し、実行可能プログラムファイルM_3の末尾に第二プロテクトコードC_2を付加することとしているが、必ずしも、このようにする必要はない。例えば、プロテクトコード付与処理部が、図６に示すように、ＬｉｂファイルL_2の末尾に、ダミーデータと第一プロテクト

コードC_1とを含んだ第一ダミーエリア①を付加し、実行可能プログラムファイルM_3の末尾に、ダミーデータと第二プロテクトコードC_2とを含んだ第二ダミーエリア②を付加するようにしてもよい。そして、２つのダミーエリア①②の合計データ量を変えないように、各ダミーエリア①②に含まれるダミーデータのデータ量をランダムに変化させるようにすれば、１つの実行可能プログラムファイルM_3を作成するごとに、その内部におけるプロテクトコードC_1,C_2のアドレスが変化するため、プロテクトコードC_1,C_2の間にある関係の解析を困難にすることができる。

【００５２】

また、以上においては、プロテクトコードチェック手続きが組み込まれたＬｉｂファイルに第一プロテクトコードをリンク前に埋め込み、実行可能プログラムファイルが完成したときに、さらに、この実行可能プログラムファイルに第二プロテクトコードを埋め込むこととしているが、かならずしも、このようにする必要はない。例えば、プロテクトコードチェック手続きが組み込まれたＬｉｂファイルにはプロテクトコードを組み込まず、実行可能プログラムファイルが完成したときに、この実行可能プログラムファイルに、プロテクトコードチェック手続きによって正当性をチェック可能なプロテクトコード(例えば、互いに関連付けられた複数のプロテクトコード)を埋め込むようにしてもよい。

【００５３】

つぎに、本実施の形態に係るプログラムファイル作成処理により作成された実行可能プログラムファイルM_3を実行することができるシステムの概略ハードウエア構成について説明する。ただし、ここでは、システムの一例として、エンタテインメント装置を挙げることとする。

【００５４】

図７は、本実施の形態に係るエンタテインメント装置７００およびその周辺機器の外観図であり、図８は、本実施の形態に係るエンタテインメント装置７００のハードウエア構成図である。

【００５５】

本実施の形態に係るエンタテインメント装置７００は、ＣＤ－ＲＯＭ、ＤＶＤ

－ＲＯＭ等の光ディスクからゲームプログラム(前述のプログラムファイル作成処理により作成された実行可能プログラムファイルM_3等)を読み出し、そのゲームプログラムにしたがってゲームを実行するものである。ここで、「ゲームの実行」とは、エンタテインメント装置７００が、ユーザ(ゲームプレイヤ)からの指示にしたがって、表示装置(ＣＲＴ、ＬＣＤ、プロジェクション装置等)上の表示映像およびオーディオ装置からの音声をゲームストーリ等にあわせて変化させることをいう。

【００５６】

図７に示すように、エンタテインメント装置７００の筐体前面部には、ユーザ(ゲームプレイヤ)がゲームを実行させるために必要となる各種の操作部が設けられている。このような操作部の具体例としては、ゲームプログラムが格納された光ディスクが装着されるディスク装着部７０１、ディスク装着部７０１からトレイを引き出すためのトレイ操作ボタン７０１Ａ、ユーザ(ゲームプレイヤ)からの入力を受け付ける各種入力受付け部(ボタン７５３Ａ,７５３Ｂ,７５４～７５８、方向キー７５２、スティック７５１Ａ,７５１Ｂ等)を有するコントローラ７５０が接続される複数(ここでは２つ)のコントローラ接続部７０２Ａ,７０２Ｂ、実行中のゲームをリセットするためのリセットボタン７０３、ゲームデータ等を記録させるためのメモリカード７３０が装着される複数(ここでは２つ)のメモリカード装着部７０４Ａ,７０４Ｂ、等が挙げられる。なお、ここで、コントローラ接続部７０２Ａ,７０２Ｂを複数設けているのは、コントローラ接続部７０２Ａ,７０２Ｂにそれぞれコントローラを接続することによって、エンタテインメント装置７００が、複数のゲームプレイヤからの指示を受け付けることができるようにするためである。

【００５７】

また、エンタテインメント装置７００の筐体背面部(前面の反対側)には、電源オンオフスイッチ(不図示)、表示装置等が接続されるＡＶ端子(不図示)が設けられている。

【００５８】

そして、このエンタテインメント装置７００の内部には、図８に示すように、

装置全体を制御するメインＣＰＵ８００、ＡＶ端子(不図示)から出力されるビデオ信号を生成するグラフィックプロセッサ(ＧＰ)８０１、コントローラ接続部７０２Ａ,７０２Ｂに接続されたコントローラ７５０およびメモリカード接続部７０４Ａ,７０４Ｂに接続されたメモリカード７３０のデータ入出力を制御するＩ／Ｏプロセッサ(ＩＯＰ)８０２、ディスク装着部７０１に装着された光ディスクからのデータ読出しを制御する光ディスク制御部８０３、ＡＶ端子(不図示)から出力されるオーディオ信号を生成するサウンド再生処理プロセッサ(ＳＰＵ)８０４、サウンドバッファ８０５、メインＣＰＵ８００およびＩＯＰ８０２が実行する各オペレーティングシステムプログラムが格納されたＯＳ－ＲＯＭ８０６、ＩＯＰのワークエリアとして用いられるＩＯＰメモリ８０７、メインＣＰＵ８００のワークエリアおよびバッファとして用いられるメインメモリ８０８、これら各部の間をつなぐバス８０９等が搭載されている。

【００５９】

このエンタテインメント装置７００に電源が投入されると、２種類のオペレーティングシステムプログラム(メインＣＰＵ用、ＩＯＰ用)がそれぞれＯＳ－ＲＯＭ８０６から読みだされ、それらがメインＣＰＵ８００とＩＯＰ８０２とによってそれぞれ起動される。これにより、オペレーティングシステムによって、装置各部の統合制御が開始され、エンタテインメント装置としての各種機能がユーザ(ゲームプレイヤ)に提供される。具体的には、光ディスクからの実行可能プログラムファイル読込みおよびその実行、コントローラを介したプレイヤからの指示受付け、プレイヤの指示に応じた映像表示および効果音･楽音出力等が可能な環境が提供される。

【００６０】

このような環境において、図９に示した光ディスクから、上述の実行可能プログラムファイル作成処理によって作成されたゲーム用実行可能プログラムファイルM_3が読み込まれると、そのゲーム用実行可能プログラムファイルM_3に定義された処理を実行するための機能構成が、メインＣＰＵ８００によってプロセスとして実現される。すなわち、図１０に示すように、エンタテインメント装置７００は、(１)実行可能プログラムファイルの所定領域(第一プログラム格納変数)の

格納データ(Ｌｉｂファイルの正当使用によるものであれば、正当な第一プロテクトコードC_1)が格納された第一プロテクトコード格納部１０００、(２)実行可能プログラムファイルの末尾領域の格納データ(Ｌｉｂファイルの正当使用によるものであれば、第二プロテクトコードC_2)が格納された第二プロテクトコード格納部１００１、(３)起動時にプロテクトコードチェック手続きを実行するプロテクトコードチェック処理部１００２、(４)プロテクトコードチェック処理部１００２からの指示にしたがって、ゲームを実行するゲーム実行処理部１００３、(５)コントローラからの入力信号をゲーム実行処理部１００３に渡す入力データ受付処理部１００４、(６)ゲーム実行処理部１００３の指示にしたがって、効果音等のオーディオ信号を生成する音声制御処理部１００５、(７)ゲーム処理部１００３の指示にしたがって、ゲーム画面映像のビデオ信号を生成する映像制御処理部１００６、が実現される。なお、プロテクトコードチェック処理部１００２には、(８)第一プロテクトコードC_1および第二プロテクトコードC_2を復号するコード復号処理部１００２ａ、(９)復号後の２つのプロテクトコードC_1,C_2の整合性をチェックするコードマッチング処理部１００２ｂ、(１０)与えられた指示にしたがって、処理を終了させる終了処理部１００２ｄ、(１１)コードマッチング処理部１００２ｂのチェック結果に応じて、終了処理部１００２ｄに処理終了指示またはゲーム処理部にゲーム実行開始指示を与える制御処理部１００２ｃ、が含まれている。

【００６１】

つぎに、図１１により、これらの機能構成部によって実行される処理について説明する。

【００６２】

光ディスクから読み込まれた実行可能プログラムファイルM_3が起動されると(Ｓ１１００)、図１０の機能構成部が以下の処理を開始する。

【００６３】

ゲームの開始前に、実行可能プログラムファイルM_3に組み込まれたＬｉｂファイルL_2が正当使用によるものか否かを判断するため、以下の自己診断処理がプロテクトコードチェック処理部１００２によって実行される。

【００６４】

まず、制御処理部１００２ｃが、コード復号の指示をコード復号処理部１００２ａに指示する。この指示に応じて、コード復号処理部１００２ａは、第一プロテクトコード格納部１０００の格納データと、第二プログラムコード格納部１００１の格納データとを取り出し、それらをそれぞれ復号する。

【００６５】

フロントエンドツールの正当使用により作成された実行可能プログラムファイルM_3であれば、１組のプロテクトコード作成に用いる関係(ここでは$C_1 = F(C_2)$)を満たすデータC_1,C_2が、このときの復号処理によって得られるはずである。そこで、コードマッチング処理部１００２ｂは、ここで得られた２つのデータがその関係を満たすか否かを診断し、その診断結果を制御処理部１１０２ｃに通知する(Ｓ１１０１)。

【００６６】

そして、制御処理部１１０２ｃは、実行可能プログラムファイルM_3に組み込まれたＬｉｂファイルL_2が正当使用によるものか否かを、コードマッチング処理部１００２ｂの診断結果に基づき判断する(Ｓ１１０２)。具体的には、１組のプロテクトコード作成に用いた関係を満たさないという診断結果が通知された場合には、プロテクトコードに矛盾ありと判断し、１組のプロテクトコード作成に用いた関係を満たすという診断結果が通知された場合には、プロテクトコードに矛盾なしと判断する。

【００６７】

以上の自己診断処理によってプロテクトコードに矛盾ありと判断した場合には、制御処理部１１０２ｃは、その旨を表すエラーメッセージを出力し(Ｓ１００３)、終了処理部１００２ｄに終了指示を与える。この指示に応じて、終了処理部１００２ｄは、所定の終了処理を実行し、プログラムを終了させる(Ｓ１１０６)。

【００６８】

一方、プロテクトコードに矛盾なしと判断した場合には、制御処理部１００２ｄは、ゲーム実行処理部１００３にゲーム実行指示を与える。この指示に応じて

、ゲーム実行処理部１００３は、ゲーム処理を開始する(Ｓ１１０４)。

【００６９】

以後、コントローラを介して送られてくるプレイヤからの指示は、入力データ受付処理部１００４からゲーム実行処理部１００３に渡される。そして、ゲーム実行処理部１００３は、入力データ受付処理部１００４から渡された指示に応じて、表示装置上の表示映像およびオーディオ装置からの音声を変化させるべく、音声制御処理部１００５と映像制御処理部１００６とを制御する。この間、ゲーム実行処理部１００３は、ゲームオーバーの条件を満たしたか否かの判断(Ｓ１１０５)を定期的に行ない、ゲームオーバーと判断したら、ゲームを終了させる(Ｓ１１０６)。

【００７０】

このように、本実施の形態に係るエンタテインメント装置によれば、読み込んだゲーム用実行可能プログラムファイルに正当なプロテクトコードが埋め込まれている場合にだけゲームを開始させ、読み込んだゲーム用実行可能プログラムファイルに正当なプロテクトコードが埋め込まれていなければ、ゲームを終了させる。したがって、フロントエンドツールの正当使用により、ゲーム用実行可能プログラムファイルが作成されていなければ、エンタテインメント装置でゲームをすることができなくなる。このため、フロントエンドツールの正当使用者以外のユーザによるＬｉｂファイル使用を止することできる。

【００７１】

なお、以上においてはエンタテインメント装置を例に挙げたが、上述の実行可能プログラムファイル作成処理により作成された実行可能プログラムファイルを実行するための装置は、必ずしもエンタテインメント装置である必要はない。例えば、プログラムの実行能力のある、一般的な電子計算機であってもよい。もちろん、実行可能プログラムファイルも、ゲームを実行させるものである必要はない。

【００７２】

また、ここでは、記憶媒体から実行可能プログラムファイルが実行装置に読み込まれるようにしているが、ネットワーク等の伝送媒体によって実行可能プログ

ラムファイルが伝送され、それが実行装置のハードディスクに格納されるようにしてもよい。

【００７３】

また、以上においては、ライブラリファイルをプロテクト対象としているが、その他のファイル(例えば、画像データ等が格納されたファイル等の、実行可能ファイル以外のファイル)をプロテクト対象としてもよい

【００７４】

【発明の効果】

本発明によれば、ライブラリ等、実行可能ファイル以外のファイルの不正コピーを防止することができる。

【図面の簡単な説明】

【図１】

本発明の実施の一形態に係る情報処理システムのハードウエア構成図である。

【図２】

本発明の実施の一形態に係る情報処理システムの機能構成図である。

【図３】

本発明の実施の一形態に係る実行可能プログラム作成処理のフローチャートである。

【図４】

本発明の実施の一形態に係る１組のプロテクトコードの生成方法を概念的に示した図である。

【図５】

本発明の実施の一形態に係る実施可能プログラムファイルが作成されるまでの処理を概念的に示した図である。

【図６】

本発明の実施の一形態に係る実施可能プログラムファイルのデータ構造を概念的に示した図である。

【図７】

本発明の実施の一形態に係るエンタテインメント装置およびその周辺機器の外

観図である。

【図８】

本発明の実施の一形態に係るエンタテインメント装置のハードウエア構成図である。

【図９】

本発明の実施の一形態に係る光ディスクの格納データのデータ構造を概念的に示した図である。

【図１０】

本発明の実施の一形態に係るエンタテインメント装置の機能構成図である。

【図１１】

本発明の実施の一形態に係る実行可能プログラム作成処理のフローチャートである。

【符号の説明】

L_1…暗号化ライブラリファイル　　L_2…ライブラリファイル

M_1…ソースファイル　　M_2…オブジェクトファイル

M_3…実行可能ファイル

１００…情報処理装置　　１０１…ＣＰＵ

１０２…グラフィックス機構　　１０３…主記憶

１０４…メモリバス制御チップ　　１０５，１０６，１０９…バス

１０７…ハードディスクドライブ　　１０８…ハードディスク

１１０…ブリッジ回路　　１１１…入力装置コントローラ

１１２…ＣＤ－ＲＯＭドライブ　　１１３…出力装置コントローラ

１２０…入力装置　　１３０…出力装置

２００…入力受付部　　２０１…起動時チェック処理部

２０２…開発成果記憶部　　２０３…コンパイル処理部

２０４…ライブラリ記憶部　　２０５…復号化処理部

２０６…一時ファイル記憶部　　２０７…リンク処理部

２０８…一時ファイル削除処理部　　２０９…コード付与処理部

２１０…制御処理部

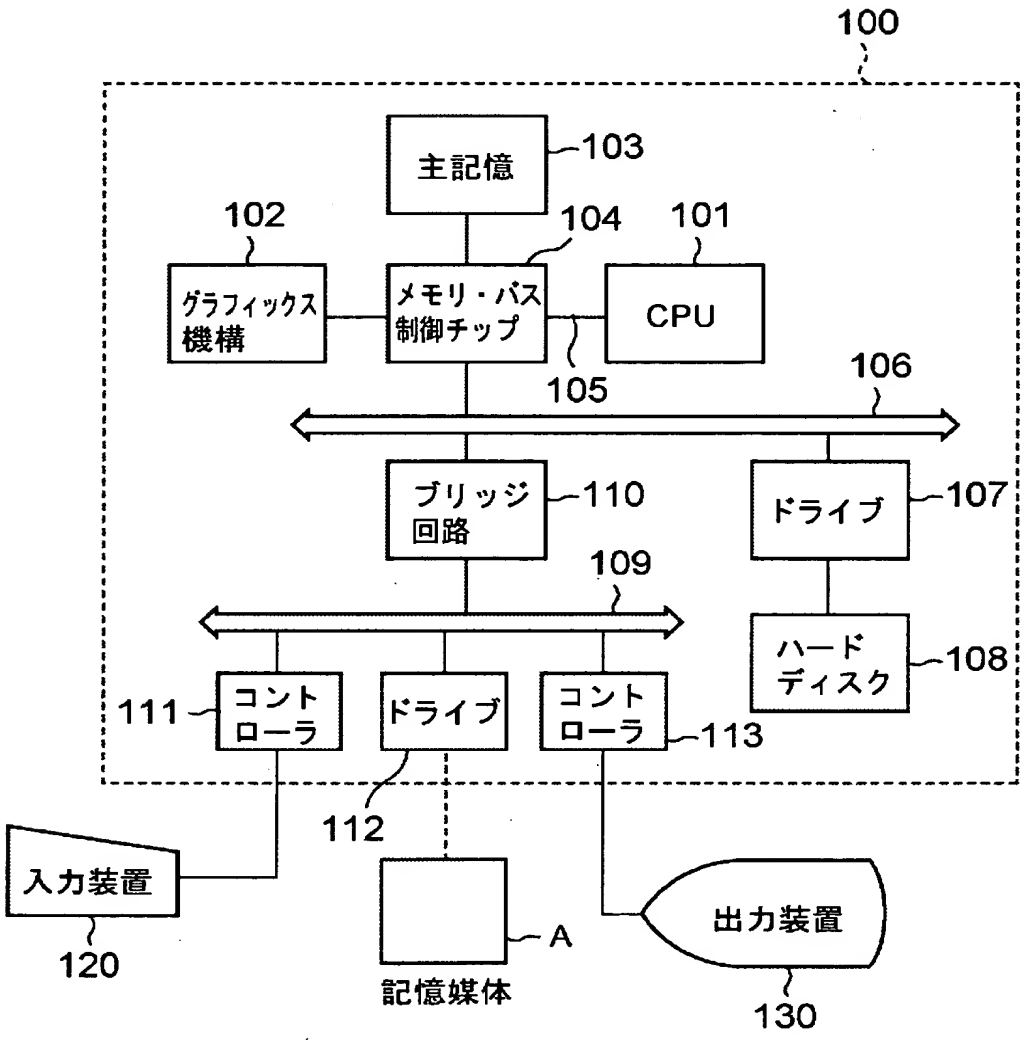
７００…エンタテインメント装置
８００…メインＣＰＵ
８０１…グラフィックプロセッサ
８０２…Ｉ／Ｏプロセッサ
８０３…光ディスク制御部
８０４…サウンド再生処理プロセッサ
８０５…サウンドバッファ
８０６…ＯＳ－ＲＯＭ
８０７…ＩＯＰメモリ
８０８…メインメモリ
８０９…バス

１０００…第一プロテクトコード格納部
１００１…第二プロテクトコード格納部
１００２…プロテクトコードチェック処理部
１００３…ゲーム実行処理部

【書類名】図面

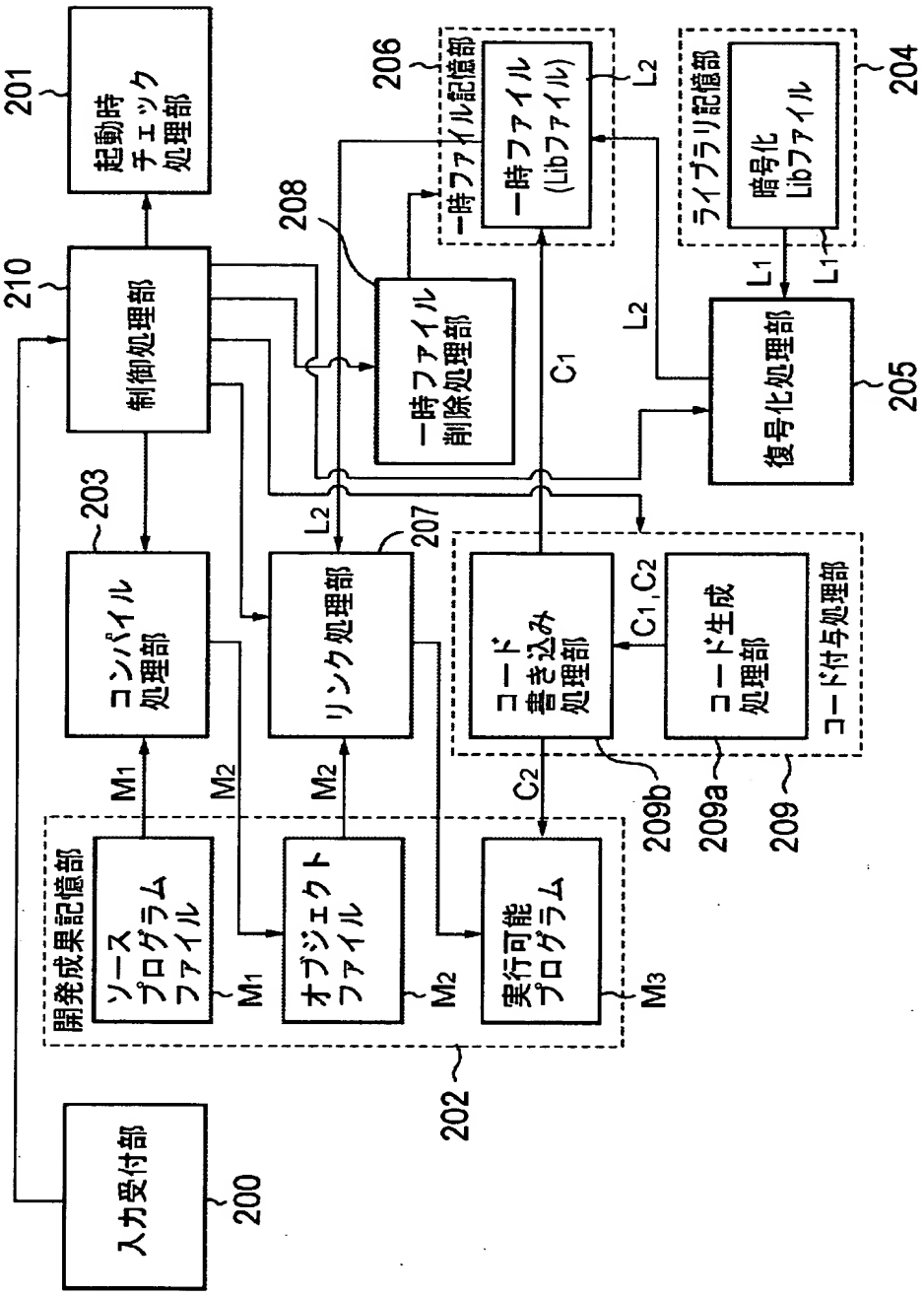
【図１】

図 1


100
主記憶
103
102
104
101
グラフィックス機構
メモリ・バス制御チップ
CPU
105
106
ブリッジ回路
110
ドライブ
107
109
ハードディスク
108
111
コントローラ
ドライブ
コントローラ
113
112
入力装置
120
A
記憶媒体
出力装置
130

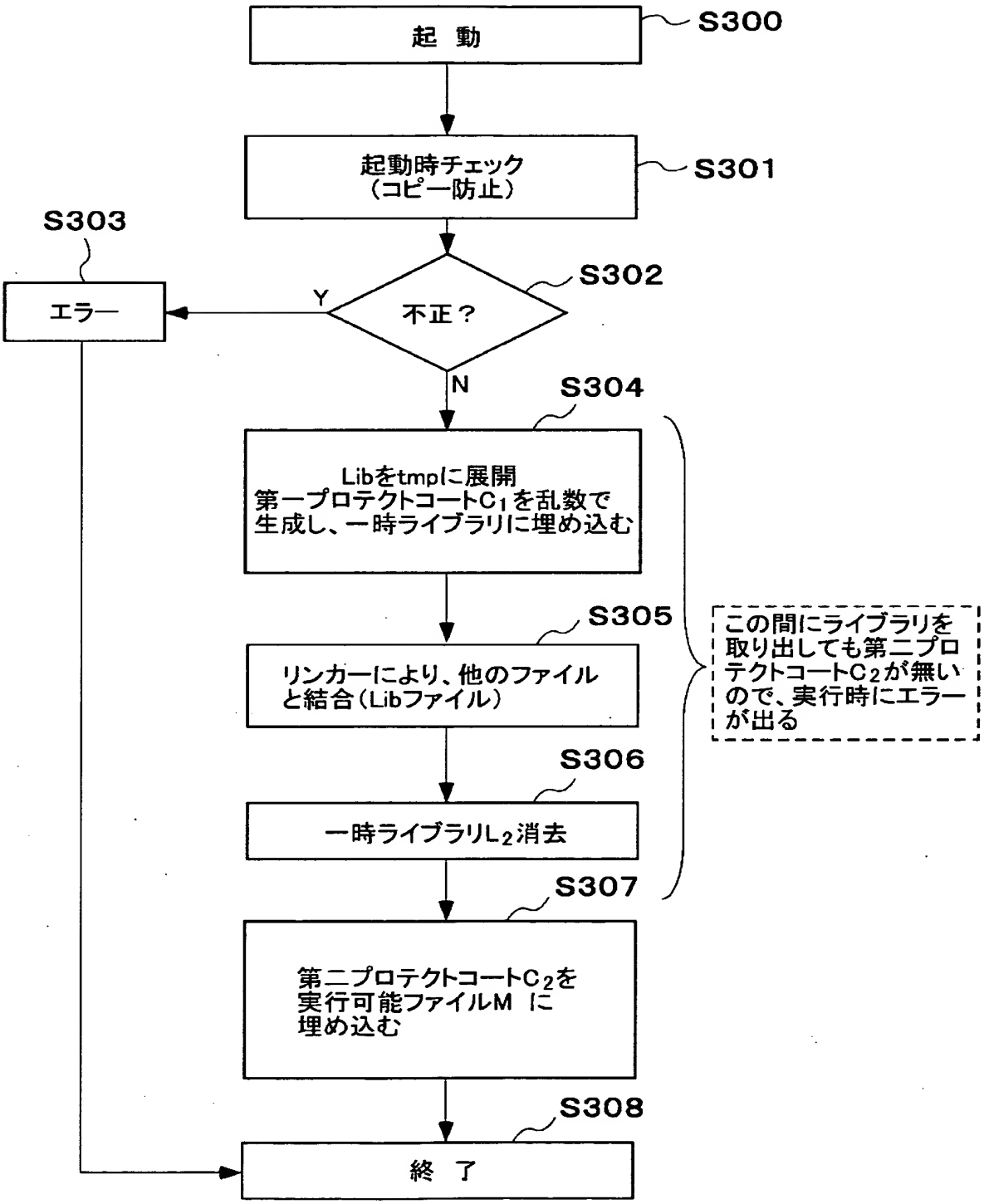
【図２】

図２


起動時チェック処理部
201
制御処理部
210
入力受付部
200
コンパイル処理部
203
リンク処理部
207
一時ファイル削除処理部
208
一時ファイル記憶部
206
一時ファイル(Libファイル)
暗号化Libファイル
ライブラリ記憶部
204
復号化処理部
205
コード書き込み処理部
コード生成処理部
コード付与処理部
209
209a
209b
開発成果記憶部
ソースプログラムファイル
オブジェクトファイル
実行可能プログラム
202
M1
M2
M3
C1
C2
L1
L2

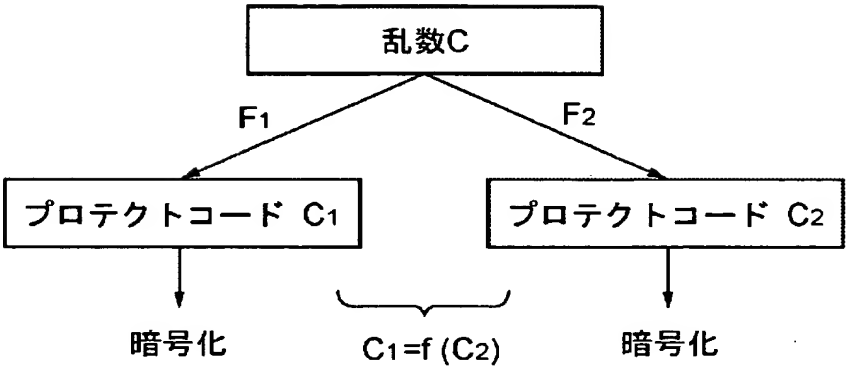
【図３】

図3


起　動
S300
起動時チェック
（コピー防止）
S301
S303
エラー
Y
不正？
S302
N
S304
Libをtmpに展開
第一プロテクトコートC1を乱数で
生成し、一時ライブラリに埋め込む
S305
リンカーにより、他のファイル
と結合（Libファイル）
S306
一時ライブラリL2消去
S307
第二プロテクトコートC2を
実行可能ファイルM に
埋め込む
S308
終　了
この間にライブラリを
取り出しても第二プロ
テクトコートC2が無い
ので、実行時にエラー
が出る

【図４】

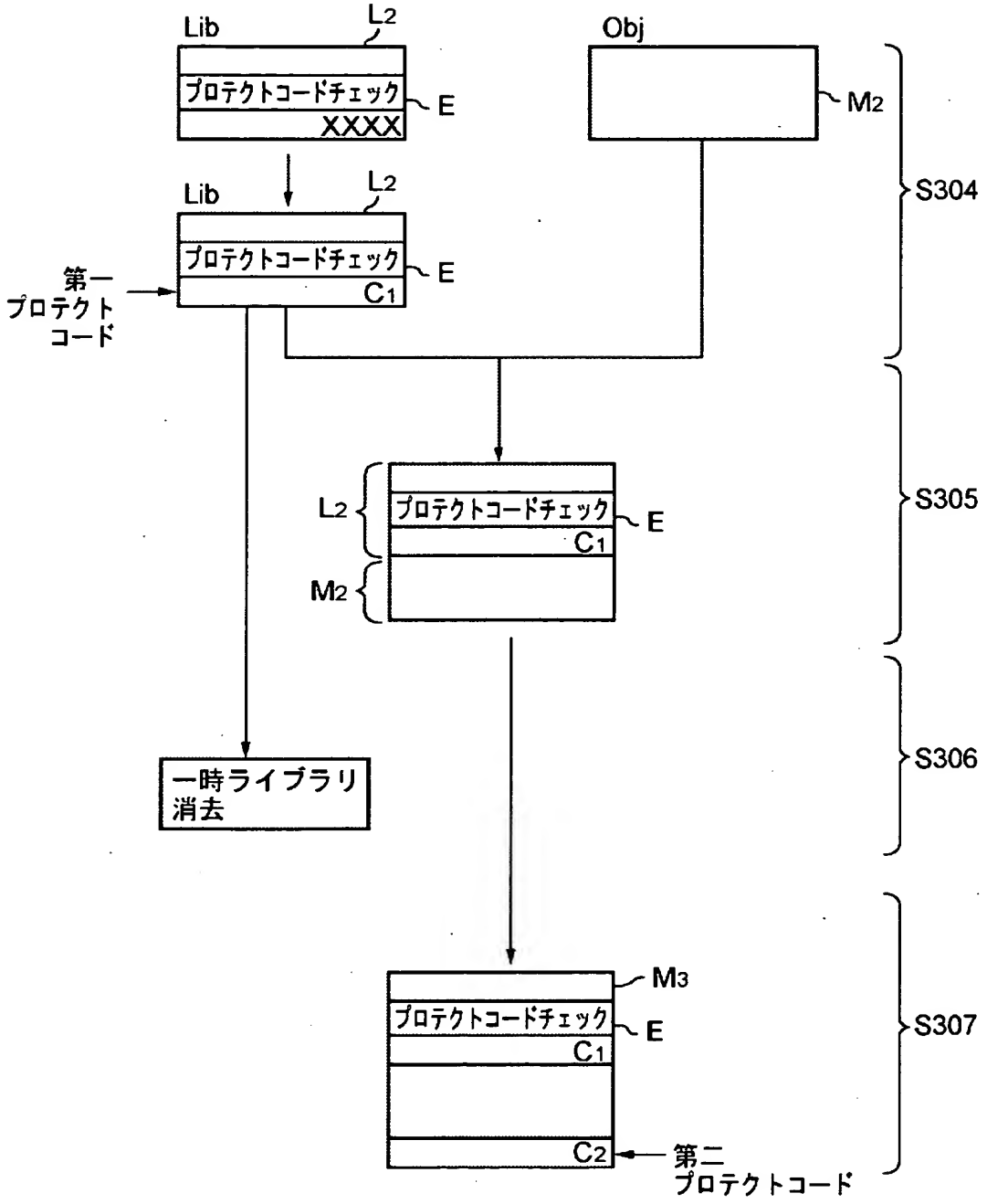
図４



出証特２００１－３０５２５７９

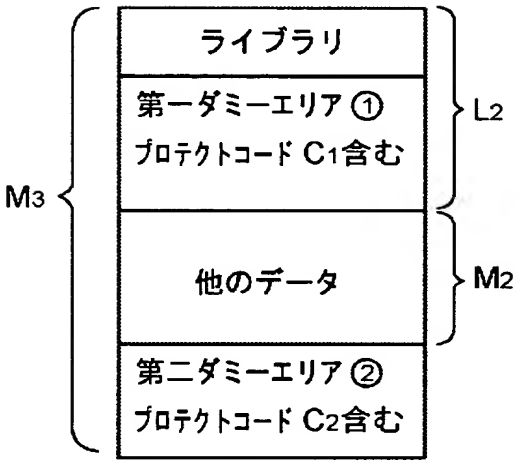
【図５】

図５


Lib
L2
プロテクトコードチェック
E
XXXX
Obj
M2
S304
Lib
L2
プロテクトコードチェック
E
C1
第一
プロテクト
コード
L2
プロテクトコードチェック
E
C1
M2
S305
S306
一時ライブラリ
消去
M3
プロテクトコードチェック
E
C1
C2
第二
プロテクトコード
S307

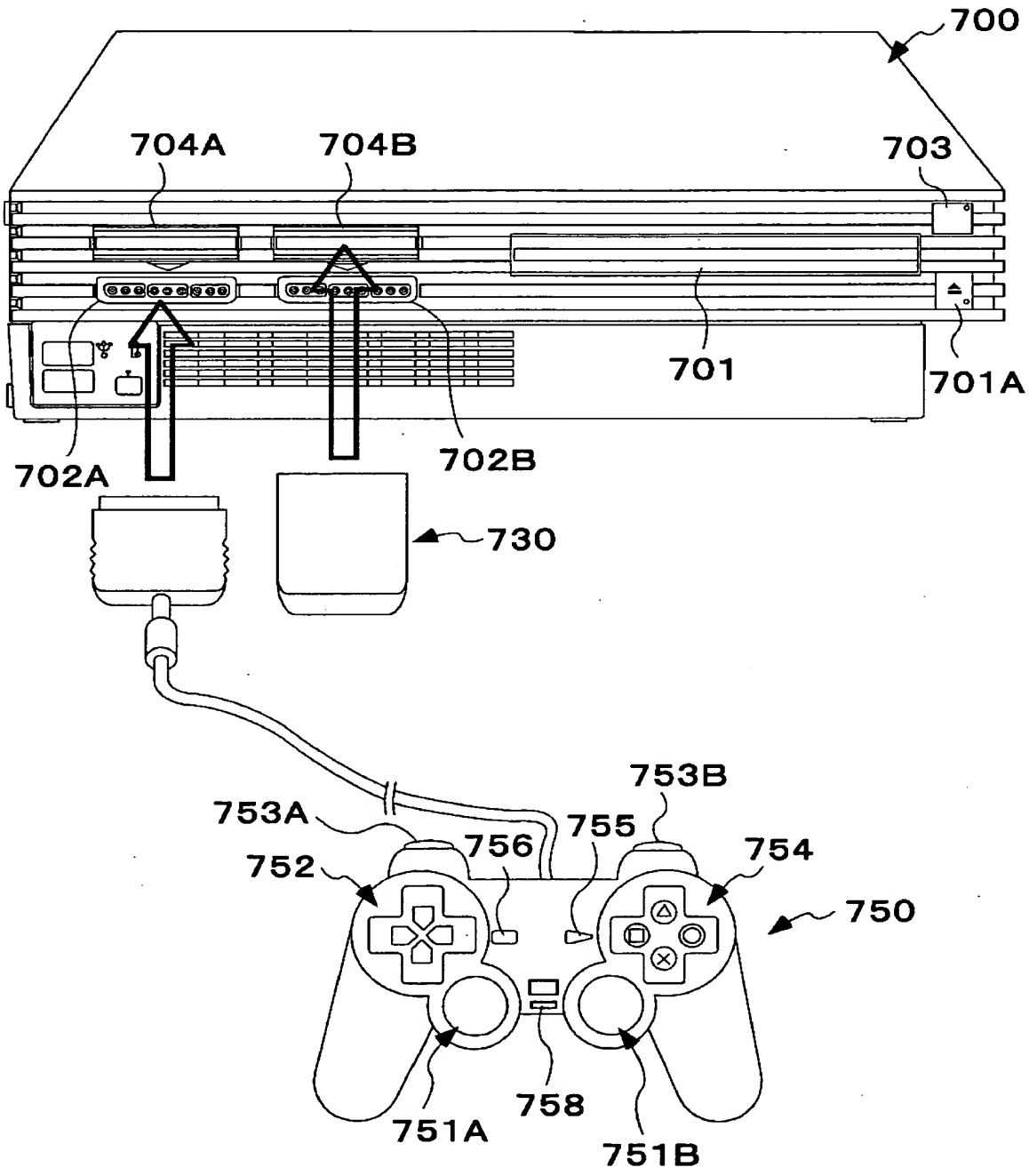
【図６】

図６


ライブラリ
第一ダミーエリア①
プロテクトコード C1含む
他のデータ
第二ダミーエリア②
プロテクトコード C2含む
M3
L2
M2

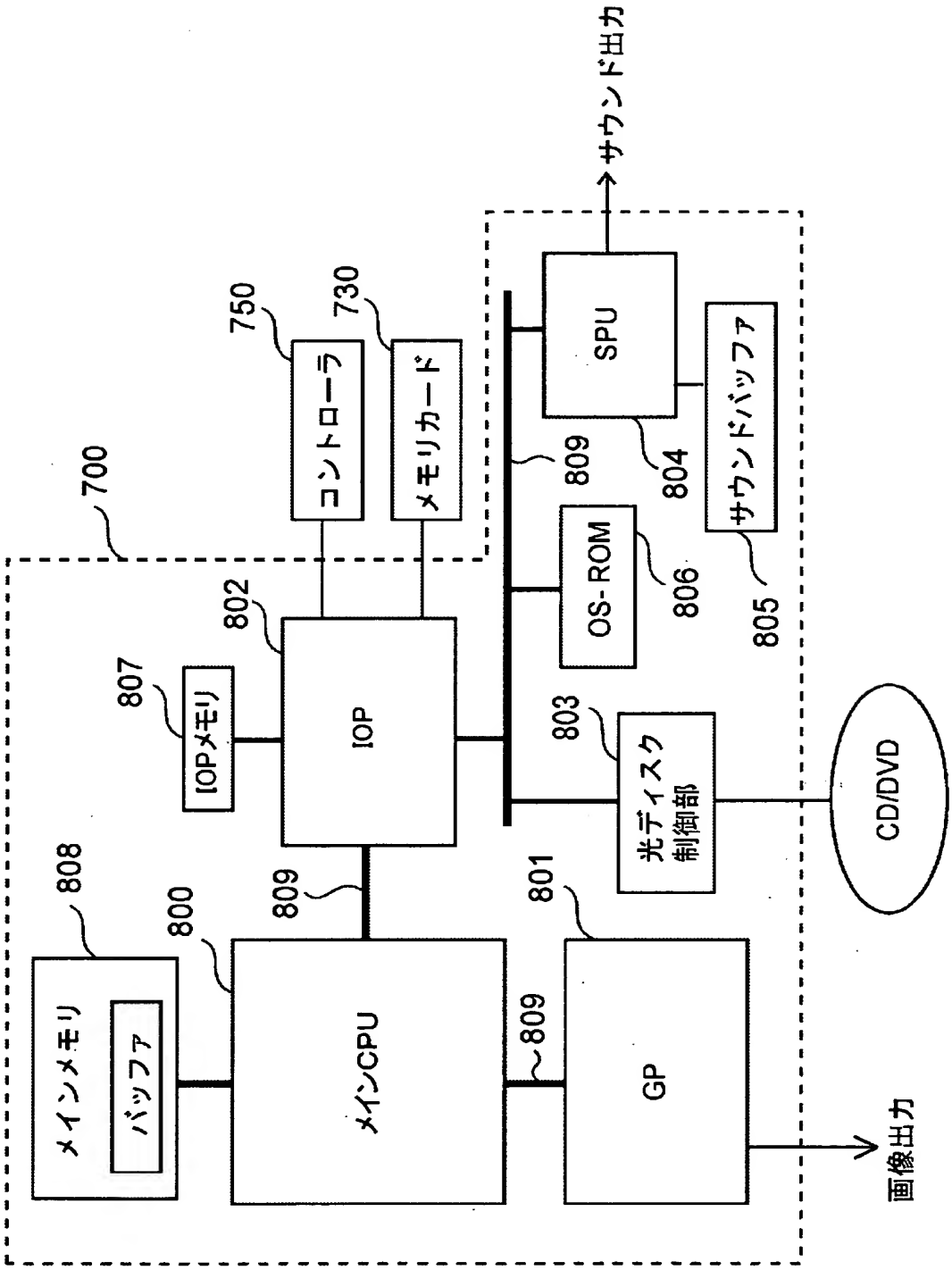
【図７】

図7


700
704A
704B
703
701
701A
702A
702B
730
753B
753A
755
756
754
752
750
751A
758
751B

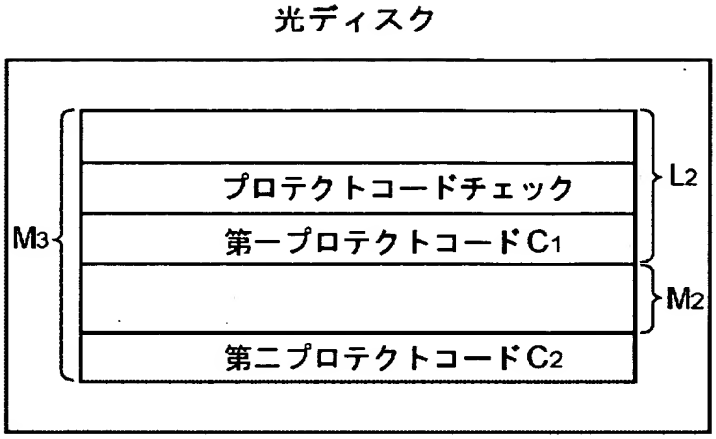
【図８】

図８


サウンド出力
750
730
コントローラ
メモリカード
SPU
サウンドバッファ
700
809
804
805
802
OS-ROM
806
807
IOPメモリ
IOP
803
光ディスク制御部
CD/DVD
808
809
801
800
メインメモリ
バッファ
メインCPU
809
GP
画像出力

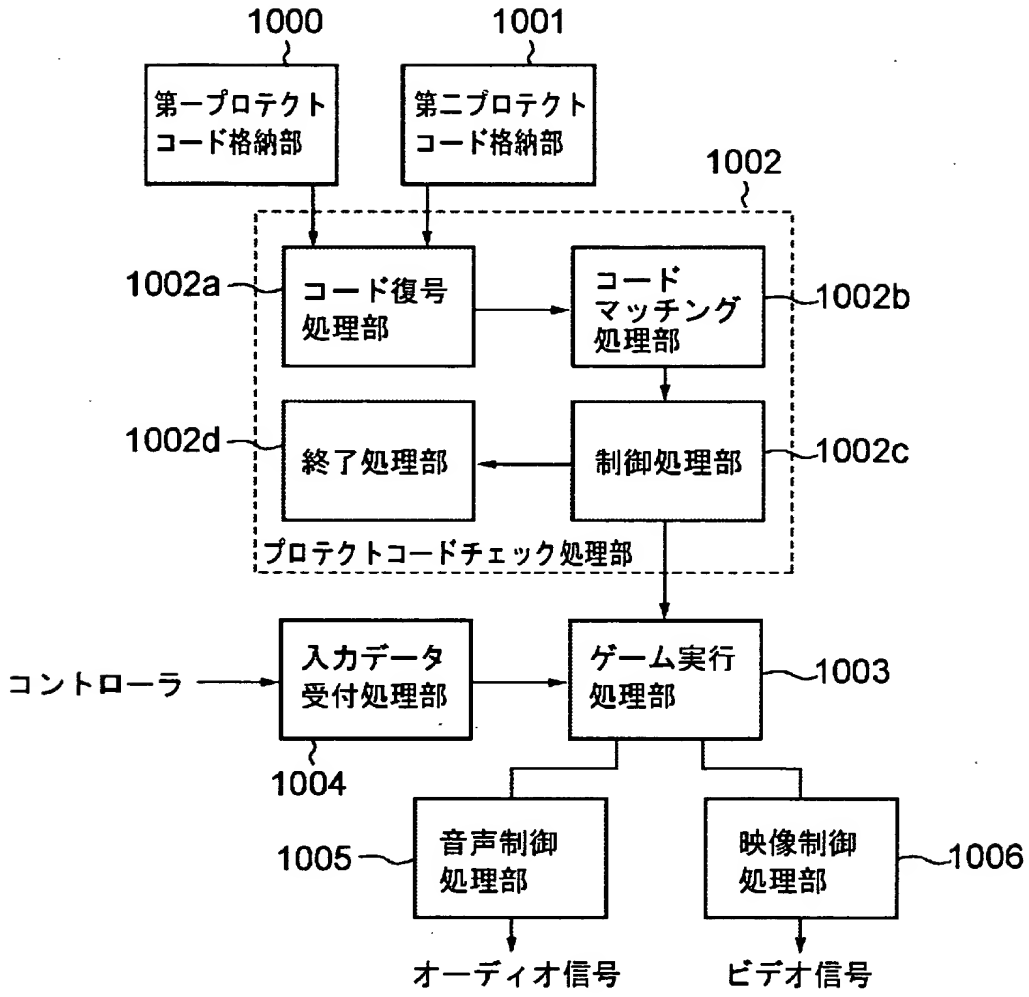
【図９】

図９



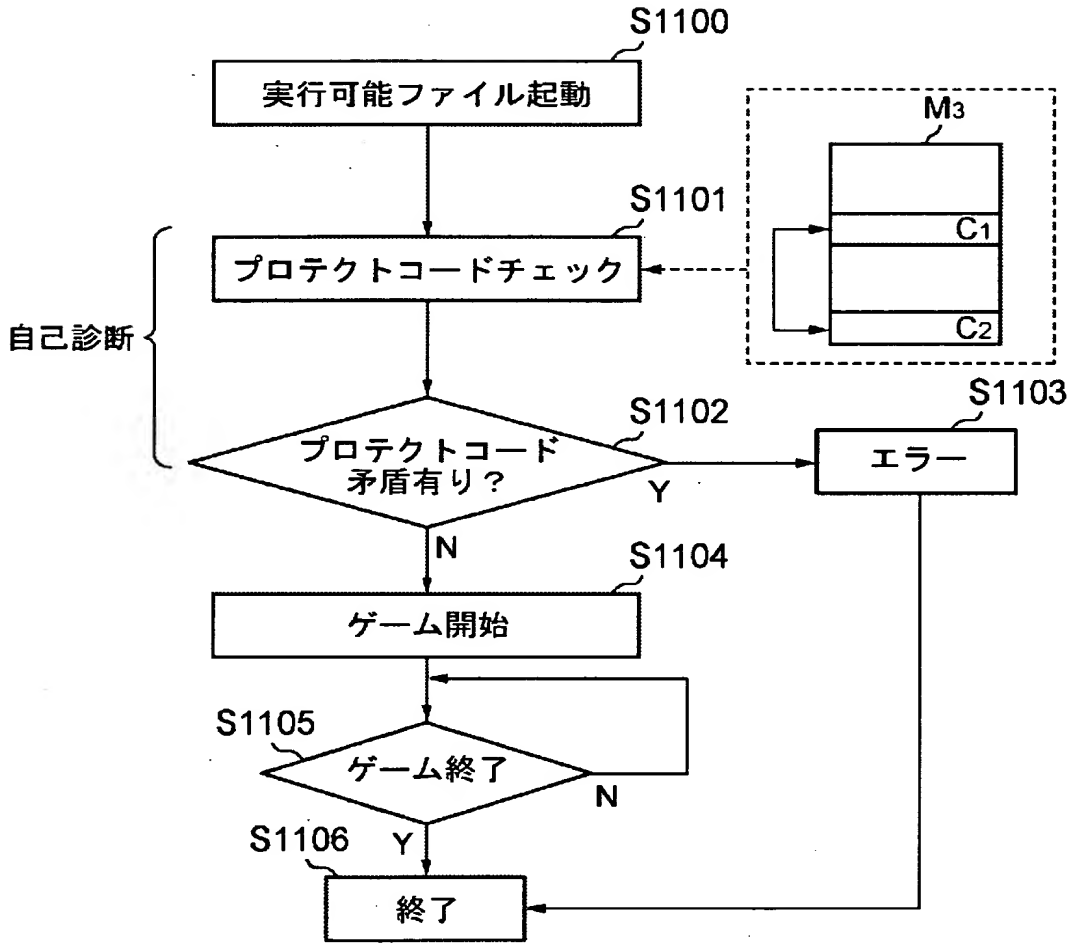
【図１０】

図 10


1000
1001
第一プロテクトコード格納部
第二プロテクトコード格納部
1002
1002a
コード復号処理部
コードマッチング処理部
1002b
1002d
終了処理部
制御処理部
1002c
プロテクトコードチェック処理部
コントローラ
入力データ受付処理部
ゲーム実行処理部
1003
1004
1005
音声制御処理部
映像制御処理部
1006
オーディオ信号
ビデオ信号

【図１１】

図 11


S1100
実行可能ファイル起動
M3
S1101
プロテクトコードチェック
C1
C2
自己診断
S1102
プロテクトコード
矛盾有り？
Y
S1103
エラー
N
S1104
ゲーム開始
S1105
ゲーム終了
N
S1106
Y
終了

【書類名】要約書

【課題】ライブラリ等の不正コピーを防止する。

【解決手段】暗号化ライブラリファイルL_2には、実行可能プログラムファイルに埋め込まれた２つのプロテクトコードに矛盾があれば、初期化段階で処理を終了させるプロテクトコードチェック手続きが含まれている。実行可能プログラムファイル作成時、暗号化ライブラリファイルL_2が復号され、それに第一プロテクトコードC_1が埋め込まれる（Ｓ３０４）。そのあと、復号後のライブラリファイルL_2とオブジェクトファイルM_2とが結合編集され、実行可能プログラムファイルが作成される（Ｓ３０５）。そして、使用済みライブラリファイルL_2を消去してから（Ｓ３０６）、実行可能プログラムファイルM_3に、第一プロテクトコードC_1と所定の関係にある第二プロテクトコードC_2がさらに埋め込まれる（Ｓ３０７）。

【選択図】図５

出証特２００１－３０５２５７９

出　願　人　履　歴　情　報

識別番号　　〔３９５０１５３１９〕

１．変更年月日　　１９９７年　３月３１日
［変更理由］　住所変更
住　所　東京都港区赤坂７－１－１
氏　名　株式会社ソニー・コンピュータエンタテインメント